Appendix

A. Proofs and Derivations

1) Proposition 1: Considering that $A = A_1^T A_1 + A_2^T A_2 + \lambda A_3^T A_3$, we examine each component separately as $A^{(1)} = A_1^T A_1$, $A^{(2)} = A_2^T A_2$, and $A^{(3)} = \lambda A_3^T A_3$, then $A = A^{(1)} + A^{(2)} + A^{(3)}$.

We select $A^{(1)} = A_1^T A_1$ as an example. As described in Section II.B, A_1 possesses the following property: for $A_1(i,j)$, if $2i - j \leq 0$ or $2i - j \geq 1$, then $A_1(i,j) = 0$. Furthermore, $A^{(1)} = A_1^T A_1$ follows from the formula $A^{(1)}(i,j) = \sum_{k=1}^{N} A_1(k,i) \cdot A_1(k,j)$. $A_1(k,i) \cdot A_1(k,j)$ can only be non-zero if $A_1(k,i) \neq 0$ and $A_1(k,j) \neq 0$. By combining these two conditions, we can deduce that $A^{(1)}(i,j) = 0$ if $|i-j| > 1$. Similarly, $A^{(2)}(i,j) = 0$ if $|i-j| > 1$ and $A^{(3)}(i,j) = 0$ if $|i-j| > 4$. Considering that $A = A^{(1)} + A^{(2)} + A^{(3)}$, then $a_{ij} = 0$ if $|i-j| > 4$ for $i,j \in [1, 2N]$.

2) Proposition 2: Similar to Proof sketch (1), we will also consider each component $A^{(1)}$, $A^{(2)}$, and $A^{(3)}$ separately. As for $A^{(1)}$,

$$A^{(1)} = A_1^T A_1 = \begin{bmatrix} 1 & 1 & & & & & \\ 1 & 1 & & & & & \\ & & 1 & 1 & & & \\ & & 1 & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & 1 \\ & & & & & 1 & 1 \end{bmatrix} \in \mathbb{R}^{2N \times 2N}.$$

As for $A^{(2)}$,

$$A^{(2)} = A_2^T A_2 = \begin{bmatrix} 0 & & & & & & \\ & 1 & & & & & \\ & & 0 & & & & \\ & & & 1 & & & \\ & & & & \ddots & & \\ & & & & & 0 & \\ & & & & & & 1 \end{bmatrix} \in \mathbb{R}^{2N \times 2N}.$$

Clearly, $A^{(1)}$ and $A^{(2)}$ satisfy $\Delta_D A^{(1)}(i,j) = 0$ and $A^{(2)}$ for $i,j \in [1, 2N]$. However, for $A^{(3)}$, the above conclusion will not apply to all data. We expand the expression $\Delta_D A^{(3)}(i,j) = 0$ as follows:

$$\begin{aligned} &\Delta_D A^{(3)}(i,j) = A^{(3)}(i+2, j+2) - A^{(3)}(i,j) \\ &= \sum_{k=1}^{N-2} A_3(k, i+2) \cdot A_3(k, j+2) - \sum_{k=1}^{N-2} A_3(k,i) \cdot A_3(k,j) \end{aligned}$$

Since $A_3(i,j) = A_3(i+1, j+2)$, $\sum_{k=1}^{N-2} A_3(k, i+2) \cdot A_3(k, j+2) = A_3(1, i+2) \cdot A_3(1, j+2) + \sum_{k=1}^{N-3} A_3(k,i) \cdot A_3(k,j)$. Considering that $\sum_{k=1}^{N-2} A_3(k,i) \cdot A_3(k,j) = \sum_{k=1}^{N-3} A_3(k,i) \cdot A_3(k,j) + A_3(N-2, i) \cdot A_3(N-2, j)$. Therefore, $\Delta_D A^{(3)}(i,j) = 0$ if and only if $A_3(N-2, i) \cdot A_3(N-2, j) = A_3(1, i+2) \cdot A_3(1, j+2)$. As shown, if $i,j \in [7, 2N-4]$, $A_3(N-2, i) = A_3(N-2, j) = A_3(1, i+2) = A_3(1, j+2) = 0$, then $A^{(3)}(i+2, j+2) - A^{(3)}(i,j) = 0$. In conclusion, $\Delta_D A^{(3)}(i,j) = 0, \quad i,j \in [7, 2N-4]$

3) Matrix Reformulation: We first substitute the equality constraints into the optimization objective, resulting in

$$\begin{aligned} \min_{\tau, s} \quad & \sum_{i=1}^{N} (\tau_i + s_i - x_i)^2 + \sum_{i=1}^{N} (s_i - v_{i \bmod M})^2 \\ & + \lambda \sum_{i=2}^{N-1} (\tau_{i+1} - 2\tau_i + \tau_{i-1})^2 \end{aligned} \tag{25}$$

We introduce $y = [\tau_1, s_1, ..., \tau_N, s_N]^T \in \mathcal{R}^{2N}$, thereby transforming the original objective function into:

$$\begin{aligned} \min_{y} \quad & \|A_1 y - x\|_2^2 + \|A_2 y - u\|_2^2 + \lambda \|A_3 y\|_2^2 \\ A_1 &= \begin{bmatrix} 1 & 1 & & & \\ & & \ddots & & \\ & & & 1 & 1 \end{bmatrix} \in \mathbb{R}^{N \times 2N}, \\ A_2 &= \begin{bmatrix} 0 & 1 & & & \\ & & \ddots & & \\ & & & 0 & 1 \end{bmatrix} \in \mathbb{R}^{N \times 2N}, \\ A_3 &= \begin{bmatrix} 1 & 0 & -2 & 0 & 1 & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & & 1 & 0 & -2 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(N-2) \times 2N}. \end{aligned} \tag{26}$$

u is a replicated expansion of the original baseline seasonal component v, i.e., $u = [v^T, \ldots v^T]^T \in \mathbb{R}^N$. In other words, it can be observed that $u_i = v_{i \bmod m}$, where m represents the period length.

We take the derivative of the objective function with respect to y and set the derivative equal to zero, yielding:

$$\frac{\partial}{\partial y} \left(\|A_1 y - x\|_2^2 + \|A_2 y - u\|_2^2 + \lambda \|A_3 y\|_2^2 \right) \tag{27}$$

$$= A_1^T (A_1 y - x) + A_2^T (A_2 y - u) + \lambda A_3^T A_3 y = 0 \tag{28}$$

Rearranging the equation results in solving $Ax = b$, where:

$$\begin{aligned} A &= A_1^T A_1 + A_2^T A_2 + \lambda A_3^T A_3 \in \mathbb{R}^{2N \times 2N}, \\ b &= A_1^T x + A_2^T u \in \mathbb{R}^{2N}, \\ u &= [v^T, \ldots v^T]^T \in \mathbb{R}^N, \\ A_1 &= \begin{bmatrix} 1 & 1 & & & \\ & & \ddots & & \\ & & & 1 & 1 \end{bmatrix} \in \mathbb{R}^{N \times 2N}, \\ A_2 &= \begin{bmatrix} 0 & 1 & & & \\ & & \ddots & & \\ & & & 0 & 1 \end{bmatrix} \in \mathbb{R}^{N \times 2N}, \\ A_3 &= \begin{bmatrix} 1 & 0 & -2 & 0 & 1 & & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\ & & & & 1 & 0 & -2 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(N-2) \times 2N}. \end{aligned}$$

B. Pre-calculation Algorithm

Algorithm 4 shows the details of OneRoundSTL pre-calculation. Line 1 constructs the vector b using the original data x and the baseline seasonal component v obtained in the cold start phase. If a missing value is encountered, Lines 4 utilize missing values handling method. Otherwise, Line 6 pre-calculates each b_i.

Algorithm 4: OneRoundSTL Pre-calculation

```
Input: $x \in \mathbb{R}^N, t \in \mathbb{R}^N, v \in \mathbb{R}^T, L \in \mathbb{R}^{2N \times 2N}, D \in \mathbb{R}^{2N \times 2N}$
Output: $z \in \mathbb{R}^{2N}$
1 construct $b \in \mathbb{R}^{2N}$ by Formula 11;
2 foreach $b_i$ do
3     if $b_i$ is missing then
4         calculate $z_i$ by Formula 20
5     else
6         calculate $z_i$ by Formula 19
7 return $z$;
```

As for time complexity, the construction and forward substitution time complexity of each b_i is O(1). Its total time complexity is O($2N$).

As for space complexity, i.e., space cost of pre-calculated results z, it is closely related to the original data, and each x_i corresponds to two z_i. Therefore, we store its corresponding x and z in the same page. The space complexity of z is O($2N$).

C. Experiments

1) Example Decomposition (continued):

2) Evaluation of Proposed Method: The evaluation of the proposed algorithm consists of two parts: (1) ablation study of the pruned LDL^T decomposition and (2) parameter study of the convergence threshold.

a) Ablation Study: Figure 13 shows the ablation study for pruned LDL^T decomposition. Among them, OneRoundSTL-RS represents LDL^T decomposition with the use of sparsity in Corollary 1 and convergence in Proposition 3, OneRoundSTL-S represents the use of sparsity only, and OneRoundSTL-O represents the straightforward LDL^T decomposition algorithm. The use of sparsity reduces the number of calculation steps and prevents the storage of zeros. Therefore, OneRoundSTL-RS and OneRoundSTL-S significantly reduce time and space costs. Convergence further reduces redundant calculations and prevents duplicate values from being stored. As a result, OneRoundSTL-RS achieves lower time and space costs compared to OneRoundSTL-S.

b) Impact of LDL^T Decomposition threshold ϵ: As ϵ decreases, the accuracy of LDL^T decomposition improves. However, Figure 14(a) shows that reducing ϵ below $1e-3$ has little effect on accuracy. In contrast, Figure 14(b) shows that decreasing ϵ will always increase the number of rows stored by L and D, resulting in a larger space cost. Thus, a moderately small ϵ balances accuracy and space cost.

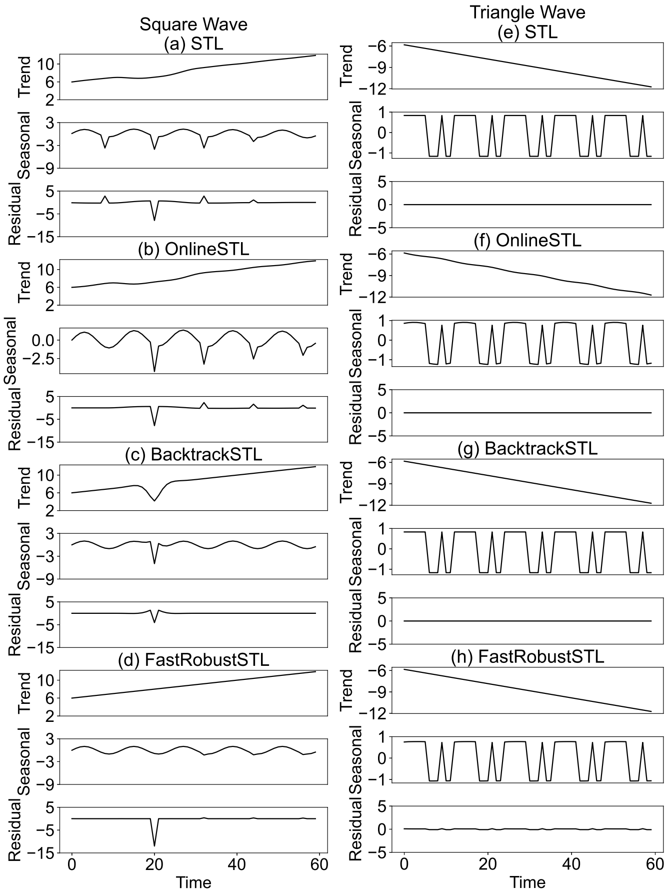

Fig. 12: Decomposition effectiveness on Synthetic Dataset (continued)

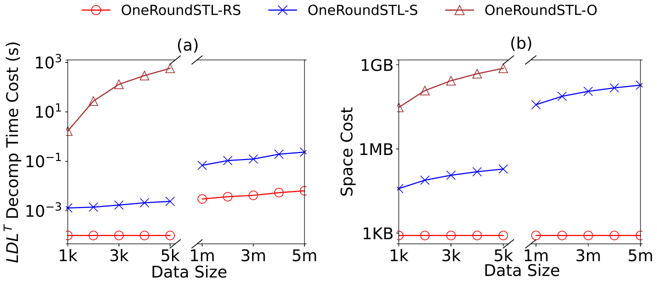

Fig. 13: Ablation study

c) Impact of Re-calculation Threshold ζ: Figure 15 shows the impact of re-calculation threshold ζ on decomposition effectiveness and efficiency. As ζ decreases, the results approach those of OneRoundSTL, indicating better accuracy. In addition, when ζ is below $1e-3$, further reduction has minimal effect on the results. The corresponding time cost, however, always increases. When ζ is 0, pre-calculated results are not used, and re-calculation is performed on the entire dataset.

3) Experiment on Varying Page Size: Figure 16 presents the query overhead w.r.t. page size. It shows that in the

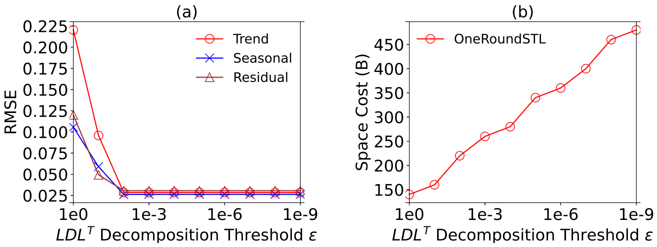

Fig. 14: Impact of LDL^T decomposition threshold ϵ

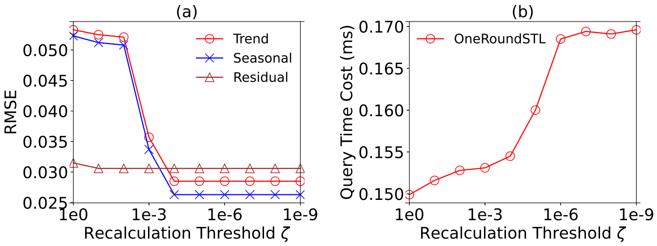

Fig. 15: Impact of re-calculation threshold ζ

query phase, a larger page size will reduce the number of concatenations. Therefore, it slightly reduces the query time cost of OneRoundSTL, e.g., in Figure 16(a). Moreover, the time cost of OneRoundSTL during query phase is consistently much lower than that of other methods.

4) *Out-of-order Evaluation:* Figure 17 shows the impact of out-of-order rate on query performance. As the out-of-order rate increases, more points require re-calculation, significantly raising the time cost for OneRoundSTL. Despite this, OneRoundSTL still outperforms OneShotSTL. The effect of out-of-order length follows a similar pattern, as illustrated in Figure 18. With a constant out-of-order rate, a larger out-of-order length results in more imputed data within the page, increasing the number of required concatenations for the query. This raises the time overhead for OneRoundSTL.

5) *Experiment on Threshold Determination:* We will present methods for determining the threshold parameters ϵ and ζ in sections 3.2 and 5. Specifically, we utilize the dataset's precision to establish these thresholds. ϵ governs the precision of L and D, which, as shown in Figure 2, undergo a squaring operation during usage. Consequently, epsilon is determined by the square root of the dataset's precision. In contrast, ζ determines the precision of the intermediate variable z, which involves only linear operations. Hence, zeta is set equal to the dataset's precision.

Figure 19 illustrates the distribution of significant figures across the dataset. We select the significant figures corresponding to the peak, which in this case represents the dataset's precision of 1e-4. Therefore, ϵ is set to 1e-2 and ζ is set to 1e-4. As demonstrated by Figures 4 and 5, this threshold selection provides OneRoundSTL with high performance and low time overhead, validating the effectiveness of our threshold determination method.

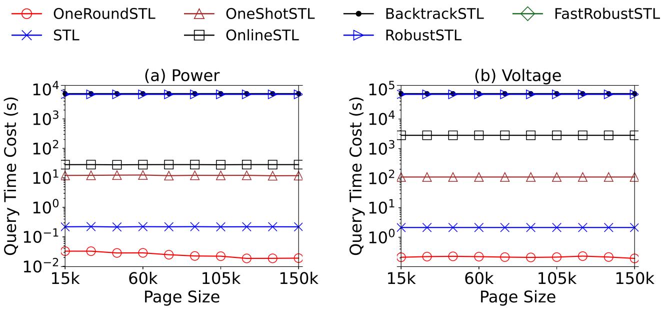

Fig. 16: Varying page size

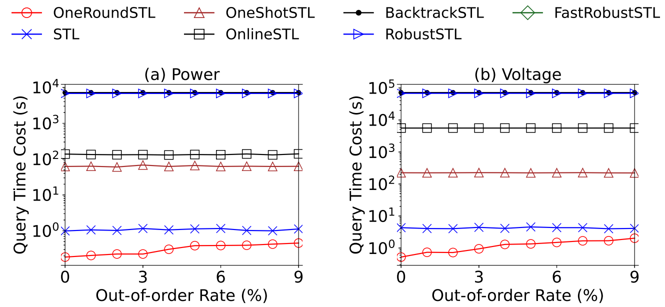

Fig. 17: Varying out-of-order rate

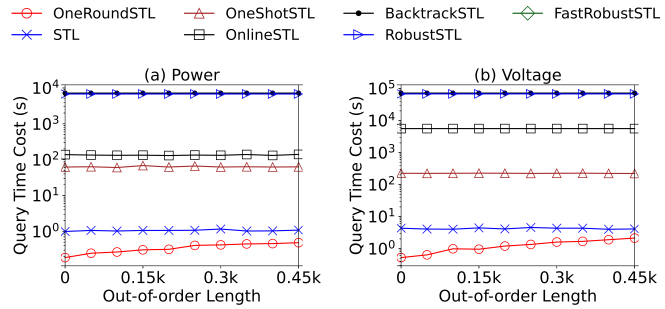

Fig. 18: Varying out-of-order length

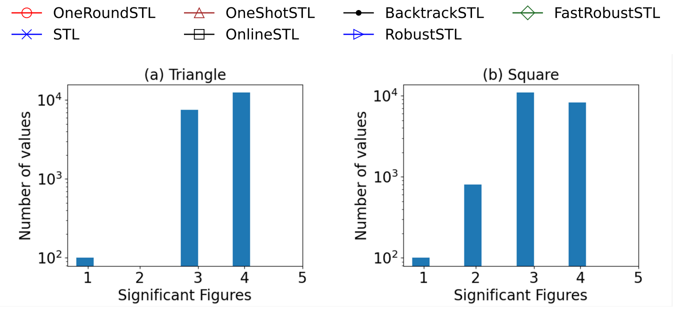

Fig. 19: Exp on Threshold Determination